

Overview



	A	B	C	D
1	Quantity	10		
2	Price	25		
3	Total	250		
4				
5	Sales Tax	16%	40	
6				
7	Total Incl. Tax		290	
8				

101

User-defined Spreadsheet



Step 1: Analyze user defined Spreadsheet
Logic and store in Application-Metafile

102



Application Metafile

103



Step 2: Produce Source Code Modules
from Application Metafile

104

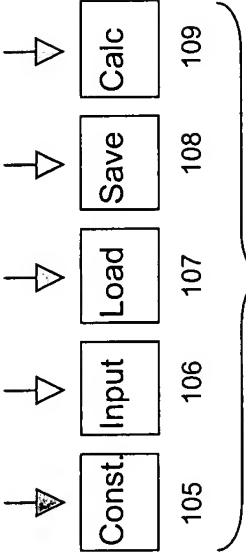


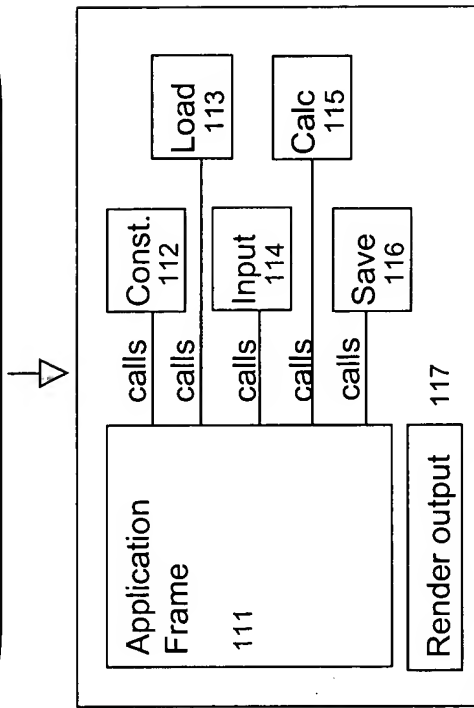
Fig. 1a

Overview (contd.)

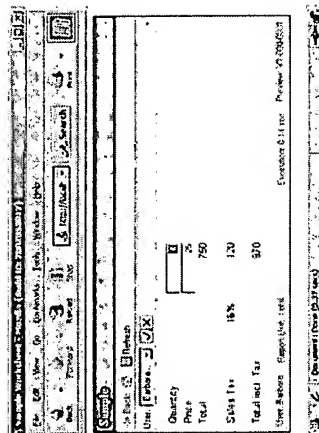
B

Step 3: Run Application Frame and call previously generated Source Code Modules

110



▽



118

Fig. 1b

Generated Internet Application



Sample Spreadsheet

	A	B	C	D
1	Quantity		10	
2	Price		25	
3	Total		250	
4				
5	Sales Tax	16%	40	
6				
7	Total incl. Tax		290	
8				

200

Fig. 2

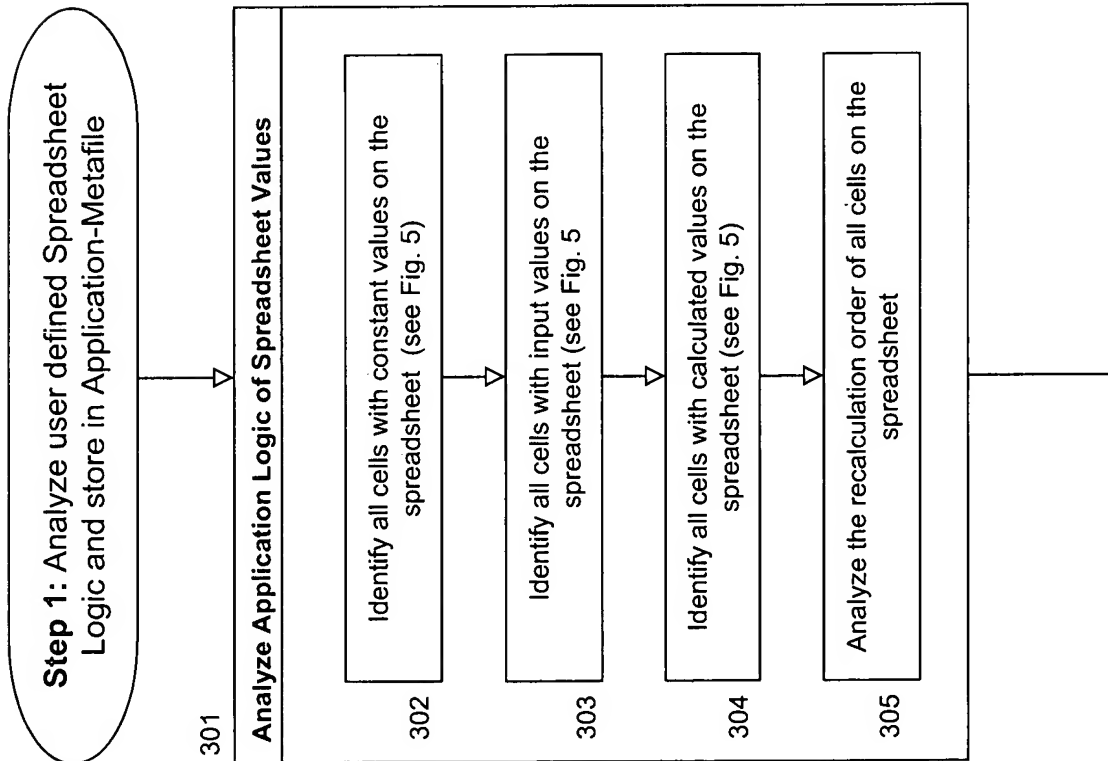


Fig. 3

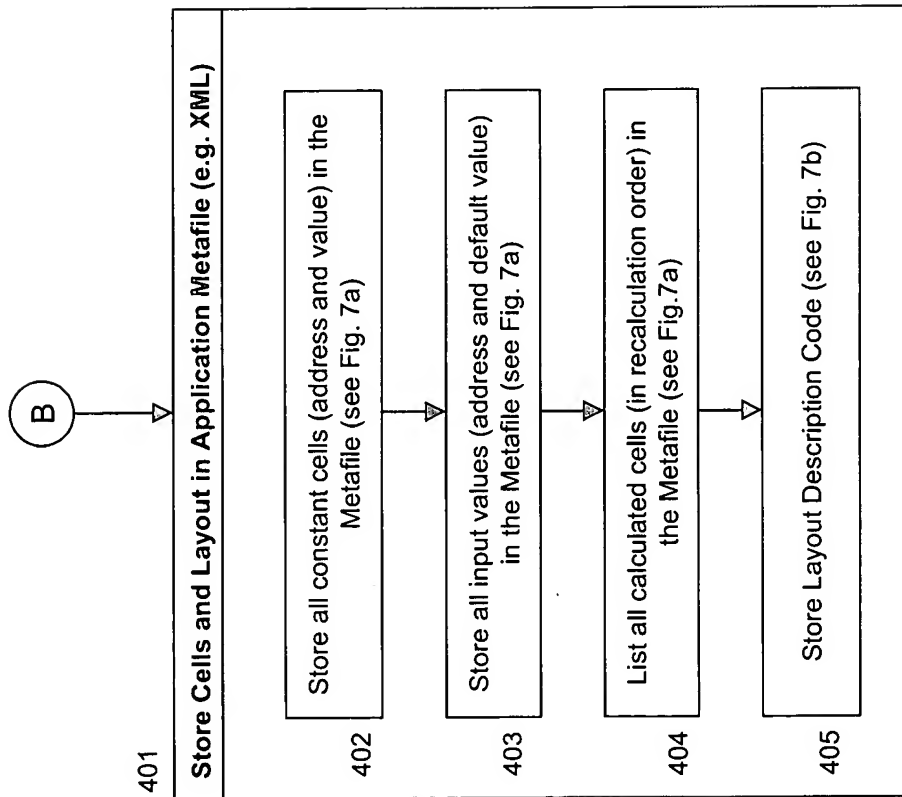


Fig. 4



Cell Types on the Sample Spreadsheet

Cells with input values

503

	A	B	C	D
1	Quantity		10	
2	Price		25	
3	Total		250	
4				
5	Sales Tax	16%	40	
6				
7	Total incl. Tax		290	
8				

501

Cells with formulas:

$C3 = C2 * C1$

$C5 = C3 * B5$

$C7 = C5 + C3$

504

Cells with constant values

502

Fig. 5



Layout Description Code for Sample Spreadsheet (HTML)

```
1 <table>
2   <col width=115>
3   <col width=40>
4   <col width=80>
5   <tr>
6     <td id='A1'></td>
7     <td></td>
8     <td id='C1'></td>
9   </tr>
10  <tr>
11    <td id='A2'></td>
12    <td></td>
13    <td id='C2'></td>
14  </tr>
15  <tr>
16    <td id='A3'></td>
17    <td></td>
18    <td id='C3'></td>
19  </tr>
20
21  ...
22
23  <tr>
24    <td id='A7'></td>
25    <td></td>
26    <td id='C7'></td>
27  </tr>
28 </table>
```

Fig. 6



Application Metafile

with Spreadsheet Logic and Layout for Sample Spreadsheet (XML)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Workbook Name="simple">
701 - <ConstCells>
    <ConstCell Name="A1">Quantity</ConstCell>
    <ConstCell Name="A2">Price</ConstCell>
    <ConstCell Name="A3">Total</ConstCell>
    <ConstCell Name="A5">Sales Tax</ConstCell>
    <ConstCell Name="B5">0.16</ConstCell>
    <ConstCell Name="A7">Total incl. Tax</ConstCell>
  </ConstCells>
702 - <InputCells>
    <InputCell Name="C1">10</InputCell>
    <InputCell Name="C2">25</InputCell>
  </InputCells>
703 - <CalcCells>
    - <CalcCell Name="C3">
      <![CDATA[ $C2*$C1 ]]>
    </CalcCell>
    - <CalcCell Name="C5">
      <![CDATA[ $C3*$B5 ]]>
    </CalcCell>
    - <CalcCell Name="C7">
      <![CDATA[ $C5+$C3 ]]>
    </CalcCell>
  </CalcCells>
+ <Worksheets>
</Workbook>
```

Fig. 7a

```
- <Worksheets>
- <Worksheet Name="Tabelle1">
704 - <SheetHTML>
  - <![CDATA[
    <table>
      <col width=115>
      <col width=40>
      <col width=80>
      <tr>
        <td id='A1'></td>
        <td></td>
        <td id='C1'></td>
      </tr>
      ...
      <tr>
        <td id='A7'></td>
        <td></td>
        <td id='C7'></td>
      </tr>
    </table>
  ]]>
</SheetHTML>
</Worksheet>
</Worksheets>
```

Fig. 7b

Step 2: Produce Source Code Modules from Application Metafile

801

Generate Code Lines for Module 'Constants' (see Fig. 11a)

Extract all cells with constant values from Metafile

802

Open a file that will hold the generated source code for processing all constant values.

803

For each cell with constant values write one line of code to disk using the appropriate syntax for the desired programming language, for example
\$A1="Quantity" in PHP

804

Close file, compile and store this module in a place where it can be executed by a calling program.

805

806

Generate Code Lines for Module 'Input' (see Fig. 11b)

Extract all cells with input values from Metafile

807

Open a file that will hold the generated source code for processing all input values.

808

For each cell with input values write one line of code to disk using the appropriate syntax for the desired programming language, for example
\$C1 = \$HTTP_POST_VARS['C1'] in PHP.

809

Close file, compile and store this module in a place where it can be executed by a calling program.

810

Fig. 8

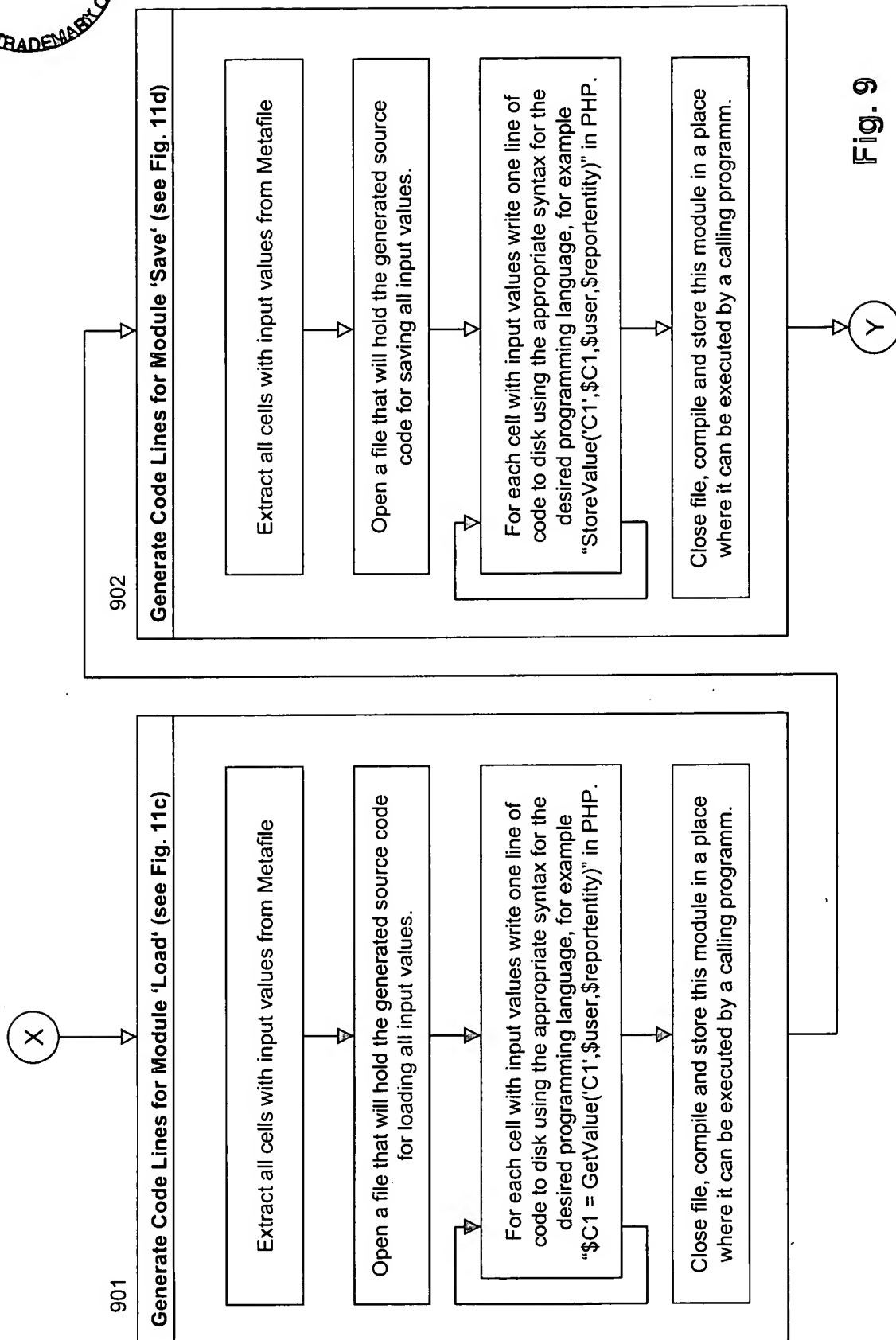


Fig. 9

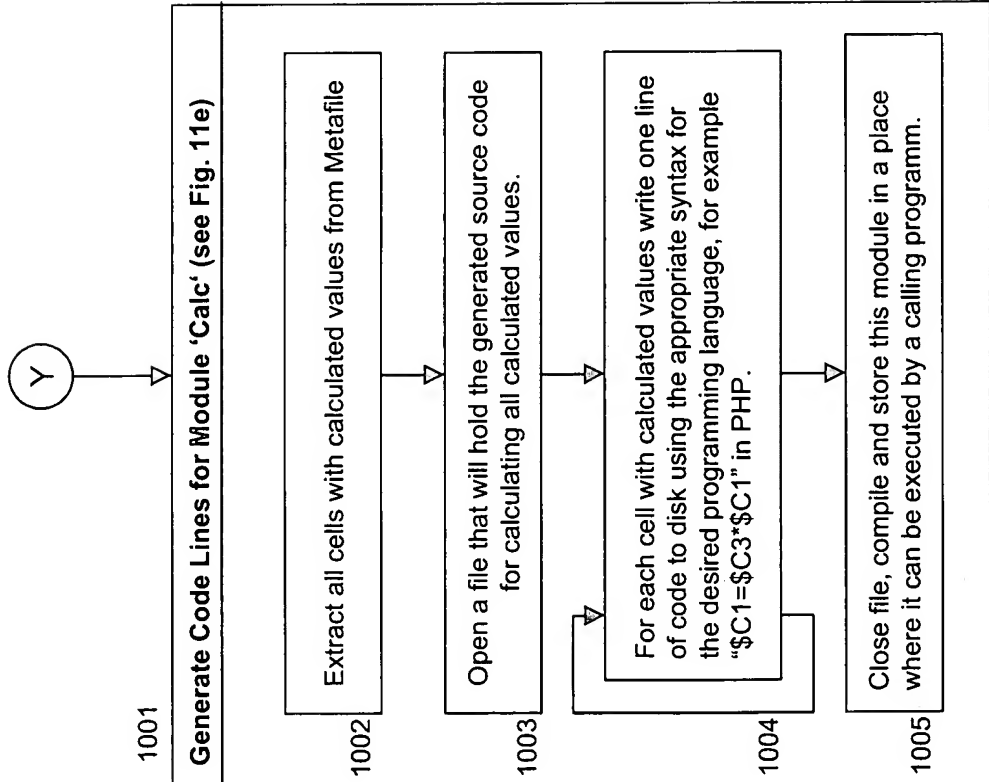


Fig. 10



Example for generated Code Modules while processing the Application Metafile from the Sample Spreadsheet

Generated Code in Module 'Constants' for the Sample Spreadsheet (HTML)

Fig. 11a

```
1 $A1 = 'Quantity';
2 $A2 = 'Price';
3 $A3 = 'Total';
4 $A5 = 'Sales Tax';
5 $B5 = 0.16;
6 $A7 = 'Total incl. Tax';
```

Generated Code in Module 'Input' for the Sample Spreadsheet (HTML)

Fig. 11b

```
1 $C1 = $HTTP_POST_VARS['C1'];
2 $C2 = $HTTP_POST_VARS['C2'];
```

Generated Code in Module 'Load' for the Sample Spreadsheet (HTML)

Fig. 11c

```
1 $C1 = GetValue('C1',$user,$reportentity);
2 $C2 = GetValue('C2',$user,$reportentity);
```

Generated Code in Module 'Save' for the Sample Spreadsheet (HTML)

Fig. 11d

```
1 StoreValue('C1',$C1,$user,$reportentity);
2 StoreValue('C2',$C2,$user,$reportentity);
```

Generated Code in Module 'Calc' for the Sample Spreadsheet (HTML)

Fig. 11e

```
1 $C3 = $C2*$C1;
2 $C5 = $C3*$B5;
```

Step 3: Run Application Frame and call previously generated Source Code Modules

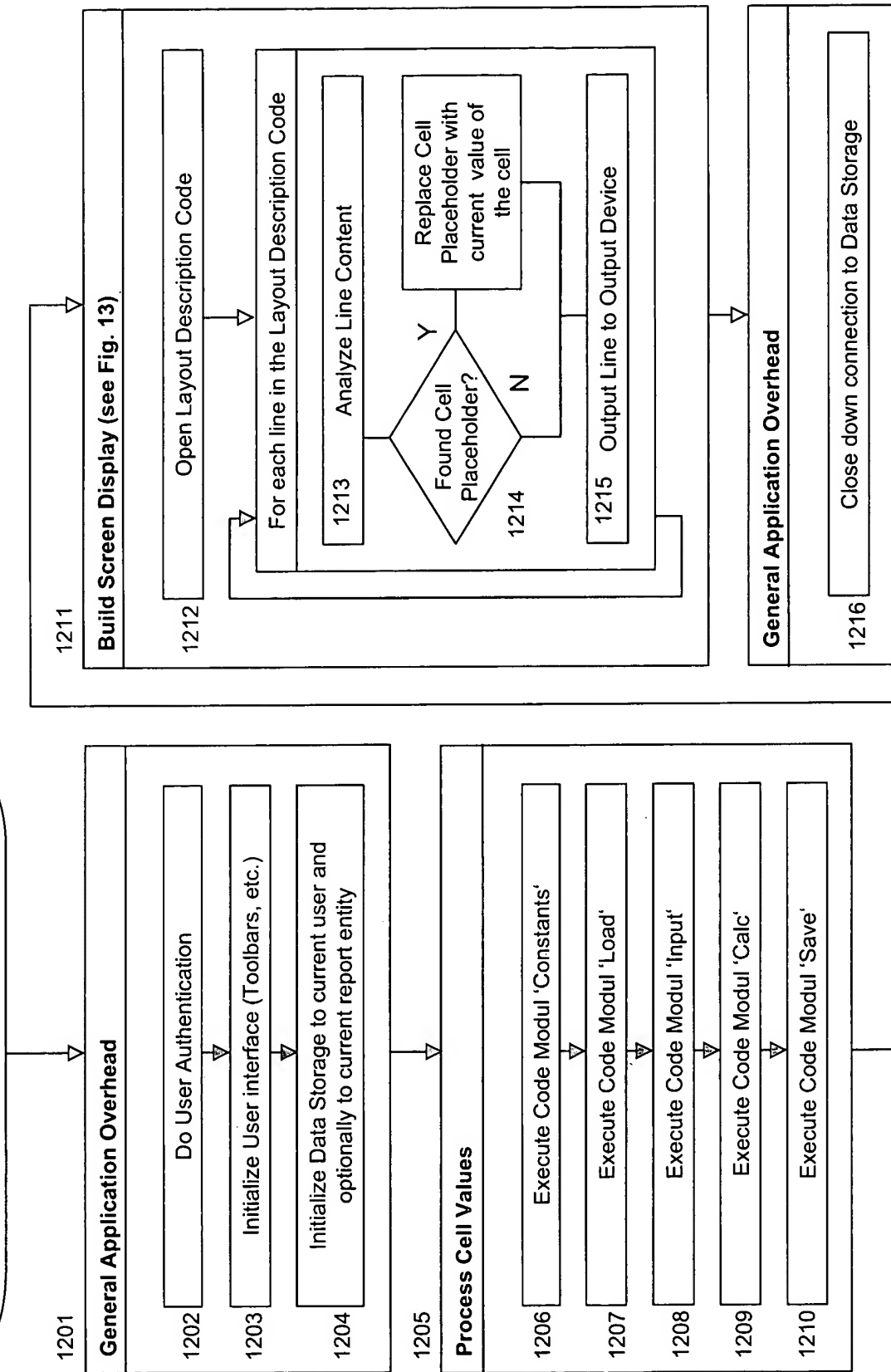


Fig. 12

Resulting Application (Running in an Internet Browser)

Sample Worksheet - Mozilla {Build ID: 2002053012}

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop

http://local-

Search

Print

Sample

Back Refresh

User: Barbara... ☒ ☒

Quantity	
Price	25
Total	750
Sales Tax	16%
Total incl. Tax	870

User: Barbara Report Unit: Total Execution: 0.14 sec. Preview: V1-000-00001

Document: Done (0:37 secs)

Fig. 13